



ECOO 2005

Programming Contest

Board wide Contest

**to be written
after March 1st**

Problem 1 - Fractorials

Marcia just learned about factorials in school. She understood that they are the product of all the integers between 1 and the number itself. For example, factorial(5) is $1 \times 2 \times 3 \times 4 \times 5 = 120$ and factorial(10) = $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 \times 10 = 3,628,800$

She decided that these numbers are too big, and wasteful. So she invented fractorials. They are the smallest possible numbers that can be evenly divided by all the whole numbers from 1 to itself. So for example fractorial(5) = 60, since $60/1$, $60/2$, $60/3$, $60/4$, $60/5$ are all whole numbers, something which is not true for any smaller number. Fractorial(10) incidentally is 2520. Something much easier to manage than 3,628,800!!!

The problem is, that they are not so easy to construct. Please help her, by writing a program that will calculate fractorials:

DATA11.txt (DATA12.txt for the second try) contain 5 integers for you to try. Write a program that will read these 5 numbers and print out the fractorials as in the sample output below: Note that no number in the input file is larger than 22 or less than 1.

Sample input:

```
4
17
22
12
1
```

Sample output:

```
Fractorial (4) = 12
Fractorial (17) = 12252240
Fractorial (22) = 232792560
Fractorial (12) = 27720
Fractorial (1) = 1
```

Problem 2 - HTTP

Simple Simon decided to invent a **Hypertext Paint Protocol**. To keep the language simple, he restricted himself to just 5 control tags:

<Un> go up **n** lines of text.

<Dn> go down **n** lines of text.

<Ln> go left **n** spaces of text.

<Rn> go right **n** spaces of text.

<Mn> repeat the last character **n** times,

in order to actually print < or > precede the character with a ?. Of course that goes for ? itself also.

For example, in order to print ">" you type ">" and if you wanted to print "?" you type "??"

for example, ?<345?><D1><L6>-<M7><D1><L6>?<678?>

would print: <345>

 <678>

DATA21.txt (DATA22.txt for the second try) contains 5 lines containing samples of Simple Simon's HTTP text: Each of the input lines contains no more than 100 characters.

Write a program that will print the text as shown in the samples output below. Note that the text will never stray more than 12 characters above or below the starting point, and never more than 39 characters to its left or right.

Under suitable input control, display each of the 5 outputs on a cleared screen or window.

Sample input:

```
?<345?><D1><L6>-<M7><D1><L6>?<678?>  
^ ^<D1><L3>0 0<D2><L4>?<<U1><R1>L<D1>--<R1>?>  
!-<M8>?<+<M3>?>-<M6>?<=<M4>?>-<M11>!  
X<D1>X<M7><U1>X<U1><L1>X<U1><L1>X<U1><L1>X<U1><L1>X<U1><L1>X<U1><L1>X<U1><L1>X<U1><L2>X<M3>  
?<M12?> means repeat 12 times<L27><U1>?<U3?> means up 3 spaces
```

Sample output:

```
<345>
-----
<678>
```

```
  ^  ^
  0  0
   L
<--->
```

```
!-----<+++>-----<===>-----!
```

```

      XXX
      X
      X
      X
      X
      X
      X
      X
X      X
XXXXXXX
```

<U3> means up 3 spaces
<M12> means repeat 12 times

Problem 3 – Binary Cipher

Consider the 32 characters made up of the six special characters: (space, period, comma, single quote, exclamation mark, question mark) and the 26 capital letters of the alphabet. They can be made to correspond with the numbers 0,1,...,31; where space = 0, period = 1, etc., question mark = 5, A = 6, B = 7, ... , Z = 31.

Messages have been encoded making use of the idea, that each of these characters can now be uniquely expressed as a 5-digit binary number.

Encoding goes as follows:

- 1 Change all the characters of a message to upper case
- 2 Change all characters other than the 32 characters above, to blanks
- 3 Translate all characters to their 5-bit binary number
- 4 Add the digits "01111" to the end of the string
- 5 Randomly add either "1" or "11" or "111" or "1111" to the start of the message,
- 6 Break the set of digits into new sets of 5, starting from the beginning of the string, and discarding the last few bits, when they form fewer than a set of 5.
Note that when you decode the string the end bits will give you a clue as to how many bits to discard from the start of the string.

DATA31.txt (DATA32.txt for the second try) contains 5 lines of coded messages.

Each line is between 1 and 250 characters long.

It is your task to write a program that will **decode** each of the 5 messages, and print them on the screen using one to three output lines for each message. Each new message must start at a new line.

Sample input:

```
SZ' H .PUKBPALCSA.NMCAFEK, .QEPXFOGDPSC'  
WQRG I'G DUGK'DK?DOEHFNABDQ EEPDRDQ ?UFK' !EDXPFW' GREDKR  
WU?T.C,XEQ?TD!C,WV?PEW' .RUBSC.  
LR KR ANFOSZ CDUI. GBFL B  
YW.MUXB W.?PLM UHQO IOPA KUHASOM.KPQPIH, .MX!PKA
```

Sample output:

```
`WHO ARE YOU?' SAID THE CATERPILLAR.  
  
THIS WAS NOT AN ENCOURAGING OPENING FOR A CONVERSATION.  
  
ALICE REPLIED, RATHER SHYLY,  
  
`I..I HARDLY KNOW, SIR,  
  
I THINK I MUST HAVE BEEN CHANGED SEVERAL TIMES'
```

Problem 4 – Three letter words

Farley Sammut was growing tired of watching kids do tic tac toe in class, and since he taught English, Farley decided to make his students do something more constructive. He challenged them to fill the tic-tac-toe board with 6 meaningful 3-letter words: Three across and three down. He told his students to make a puzzle out of this: Create a list of 9 words, and challenge each other to pick the six that would fit in the square.

Here was Farley's example:

BOB, SAM, TOM, ORA, ART, BOA, ORR, BAT, MAL
Discarding SAM, TOM and MAL:

B	O	B
O	R	A
A	R	T

Unfortunately, not all students could find 6 meaningful words that would work this way. DATA41.txt (DATA42.txt for the second try) contains 5 lines of 27 letters. The 27 letters represent 9 three-letter words.

Write a program that will help out Mr. Sammut: He needs a program that will read in 9 three-letter words and that would find the (up to) six words that would solve the puzzle in the best possible way. Your output should look like the example below: For each of the 5 sets, print out not only the words of the puzzle, but also how many of the 9 words are used in the puzzle. You may assume that the best solution would always include 3 horizontal words.

Sample input (only three of five examples are shown)

```
ARTBOAORRBATSATRATSONBOBORA  
TORINAKISONETWOESTMATREXXIN  
BOOPALODOSAMPOTADDMICMACLOT
```

Sample output

```
h1 = BOB          v1 = BOA  
h2 = ORA          v2 = ORR  
h3 = ART          v3 = BAT  
  
6 words used  
  
h1 = TOR  
h2 = INA          v2 = ONE  
h3 = REX  
  
4 words used  
  
h1 = POT          v1 = PAL  
h2 = ADD          v2 = ODO  
h3 = LOT  
  
5 words used
```



ECOO 2005

Programming Contest

Regional Contest

April 9 2005

Problem 1 - Scrabble

This is a variation on the well-known board game SCRABBLE:

You are given a word, composed of letters, each with their own value (identified by the subscript number beside):

A_1 B_3 C_3 D_2 E_1 F_4 G_2 H_4 I_1 J_8 K_5 L_1 M_3
 N_1 O_1 P_3 Q_{10} R_1 S_1 T_1 U_1 V_4 W_4 X_8 Y_4 Z_{10}

A word may be placed horizontally (from left to right) or vertically (from top to bottom). The score for a word is the product of the square multiplied by the value of the letter. For instance, if the word "PENNY" were to be placed horizontally starting in the top left square, its value would be $(2 \times 3) + (1 \times 1) + (3 \times 1) + (2 \times 1) + (3 \times 4) = 24$. The challenge is to find the place on the board where the word will have the **greatest** value, given this method of finding the score.

```

2 1 3 2 3 1 3 1 2 1 2 1 4 1 4
2 2 1 2 2 2 1 5 3 2 2 1 3 1 2
1 2 4 2 3 1 1 3 1 3 1 1 2 2 3
1 2 2 5 1 5 4 1 1 1 1 1 1 3 3
1 3 1 1 2 1 2 3 4 2 3 3 3 3 3
2 1 1 1 2 2 1 1 2 4 3 2 1 1 3
4 3 1 2 1 1 2 3 3 4 1 1 1 1 1
5 1 4 4 2 1 2 1 2 1 5 2 3 1 3
5 4 5 1 1 5 1 2 1 2 1 2 5 3 1
1 3 1 2 3 1 3 2 3 2 3 3 1 4 2
1 1 3 3 2 1 1 2 1 1 4 3 4 1 2
1 1 5 2 4 4 1 2 3 1 3 2 2 3 5
2 3 2 1 5 2 1 4 3 4 1 4 2 2 2
1 1 1 2 1 4 3 5 1 1 2 3 1 3 3
3 1 4 2 1 3 1 1 5 4 2 1 1 3 3

```

DATA11.txt (DATA12.txt for the second try) contains 15 lines of 15 digits followed by 5 words in upper case letters. The 15 lines of 15 digits represent the board, on which you must place the 5 words. The output will be the maximum value that each word could earn on this board. You may assume that each word starts with a clean board, i.e. a board with no previous words on it. In the sample output below, for example, QUARTER is worth 67, placed horizontally, starting at (1,9); LOONIE too, attains its highest value horizontally starting at (1,9).

Sample input:

```

213231312121414
221222153221312
124231131311223
122515411111133
131121234233333
211122112432113
431211233411111
514421212152313
545115121212531
131231323233142
113321121143412
115244123132235
232152143414222
111214351123133
314213115421133
QUARTER
PENNY
DIME
NICKEL
LOONIE

```

Sample output:

```

QUARTER is worth 67 points
PENNY is worth 41 points
DIME is worth 30 points
NICKEL is worth 48 points
LOONIE is worth 21 points

```


Problem 2 - Partitions

Ramanujan was a brilliant mathematician, born in poverty in southern India. He had little formal training in mathematics, and yet he spent the years between 1903 and 1913 cramming notebooks with page after page of mathematical formulas and relationships that he had uncovered. He died in 1920, at the age of 32. Ramanujan investigated in particular the properties of large numbers, and one of his investigations introduced the notion of partitions:

The number 5 can be divided in several ways, and expressed as sums of its various parts:

$$5 = 1+1+1+1+1$$

$$5 = 1+1+1+2$$

$$5 = 1+2+2$$

$$5 = 1+1+3$$

$$5 = 2+3$$

$$5 = 1+4$$

$$5 = 5$$

Partition(5) = 7 because there are 7 different ways (different arrangements of the same integers do not count) in which the number 5 can be partitioned. In the same way, it may be found that partition(10) = 42 and partition(50) = 204226

DATA21.txt (DATA22.txt for the second try) contains 5 integers. Write a program that will read these 5 numbers and print out the partitions as in the sample output below. Note that no number in the input files is larger than 50 or less than 1.

Sample input:

```
5
50
22
12
1
```

Sample output:

```
Partition(5) = 7
Partition(50) = 204226
Partition(22) = 1002
Partition(12) = 77
Partition(1) = 1
```

Problem 3 – XO life

A family of Exopods (X's) and Ozopods (O's) are living in a digital world. Each individual lives all its life in one location on a 20x20 grid. Even though the X's and O's never move, they do communicate with any one of their up to 8 adjacent neighbours: Their daily routine is composed of three consecutive events: Dying, Birthing and Changing.

- 1 An individual, whether Ozopod or Exopod, will die from overcrowding, when it finds that it has more than 5 neighbours: This happens during the Dying time, the first event of the day.
- 2 If an empty location has exactly 3 Exopod neighbours, a new Exopod is born there, and if an empty location has exactly 3 Ozopod neighbours, a new Ozopod is born there. This happens during the Birthing time, the second event of the day.
(Occasionally an empty location has 6 neighbours: 3 Ozopods and 3 Exopods. When this happens, an Ozopod will be born there)
- 3 Any Exopod that has at least one Ozopod neighbour will become an Ozopod, and any Ozopod that has at least one Exopod neighbour will turn into an Exopod. This happens during the Changing time, which marks the end of the day's cycle.

DATA31.txt (DATA32.txt for the second try) contains 5 sets of data. The first line of each set contains two numbers: **n**, between 5 and 20, representing the number of individuals in the XO world, and **d**, between 1 and 20, representing the number of days that must elapse for a display of the world. The next n lines contain 3 values: a single letter followed by two numbers (each separated by a space). The letter is either a capital X or a capital O, representing the kind of individual. The two numbers are integers between 1 and 20, which represent the x and y coordinates of that individual.

You program should display each of the 5 worlds in the form of a 20x20 set of characters, where (1,1) is located at the **bottom left** corner and (1,20) at the **top left**. Represent the Exopods by the letter X and the Ozopods by the letter O. Any unoccupied space should be displayed as a period (“.”). Advancing from one of the 5 worlds to the next one should be accomplished by a suitable signal from the keyboard or mouse (you do not need to show the world after each day, only after the given number of days has elapsed).

Problem 4 – Billiards

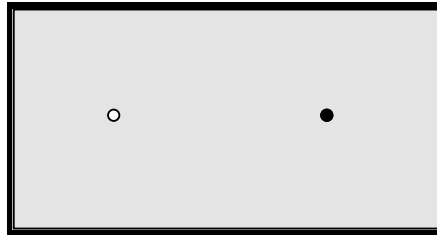
A pool table has dimensions 200 cm long by 100 cm wide and there are no pockets (it is in fact a billiards table). One end of the table faces east and the other end faces west. Two balls, a red and a white ball are located on the east-west axis, the center of the white ball 50 cm from the west bank and the center of the red ball 50 cm from the east bank of the table. Each ball has a 3 cm radius. The north bank has measurements on them: at the west end it marks 0 cm, in the middle of the bank is measures 100 cm and at the east end it marks 200 cm.

The white ball is shot so that it bounces off the north bank first and may or may not eventually hit the red ball. You may assume that:

- the white ball is hit with enough force so that it can bounce off 7 banks or the red ball, whichever comes first
- each time it bounces, the angle of reflection of the ball is always equal to the angle of incidence
- the two balls hit when their centres are 6 cm apart.

Write a program that will count how many bounces the white ball makes before it hits the red ball, if at all, given where it first bounces off the north bank.

DATA41.txt (DATA42.txt for the second try) contains 5 numbers between 3 and 197 representing the place where the ball first bounces off the north bank.



Sample input

```
69
100
75
90
3
```

Sample output

```
The white ball hits the red after 6 banks
The white ball hits the red after 1 banks
The white ball hits the red after 2 banks
The white ball does not hit the red
The white ball hits the red after 3 banks
```



ECOO 2005

Programming Contest

Ontario
Finals

May 14, 2005

Problem 1 – Longest Chain

Imagine a rectangle composed of upper case letters. In it are chains of various length. A chain is defined as a succession of distinct letters of the same kind. Each letter of the chain has a unique follower letter, and a unique predecessor. The first letter of the chain of course does not have a predecessor, and the last letter of the chain does not have a follower. Any follower may be above, below, to the left or to the right of the letter it follows. No letter occurs more than once in the chain. In the example below, the cluster of 8 L's found at the bottom of the rectangle contains several L-chains, depending on one's point of view. However, the largest L-chain one can form only has 7 letters. Note also, that the l-chain on the left is either of length 7 or 8, depending where you start. It is your task to find the longest chain and identify the letter that forms the chain.

```
YOHQGGQNEQDDJPT
JMSPXGPXKICLVQW
MSSHLGRVKBBSOSN
XIUTGGBKTEFQQQQ
CIFIBNBLBVEXCIW
IIILBBBHGKRMJOM
IEIBRLATFZQXBYY
IDIPALLLLEIRUSM
IVULQLLLKGJSWON
```

The first line of DATA11.txt (DATA12.txt for the second try) contains five sets of data. Each set contains two integers on separate lines, n (the length) and m (the width), followed by n lines of m characters, where $2 < n < 41$ and $2 < m < 41$.

In each set, find the largest chain of letters, and print the resulting data on the screen as shown in the sample below. If there is a tie, any of the (longest) solutions will be valid.

Sample input: (3 sets of 5 only)

```
10
15
GTGTAUJEPJOZIIM
YOHQGGQNEQDDJPT
JMSPXGPXKICLVQW
MKSHLGRVKBBSOSN
XIUTGGBKTEFQQQQ
CIFIBNBLBVEXCIW
IIILBBBHGKRMJOM
IEIBRLATFZQXBYY
IDIPALLLLEIRUSM
IVULQLLLKGJSWON
3
5
AAABB
BBBAB
BABBB
4
9
AAABAAAA
BAAABABAB
BAAAAABAB
BABBBAAAB
```

Sample output:

```
I has a chain of 8 characters
B has a chain of 10 characters
A has a chain of 19 characters
```

Problem 2 – Link5 Cipher

The **Link5 Cipher** is fiendishly hard to crack since it is based on relatively large numbers. The following are the steps that would be followed to encode the message:

"THERE IS NO PLACE LIKE HOME" using the **Link5 Cipher**

(1) Add as many spaces as is necessary to make the number of characters a multiple of 5:

"THERE IS NO PLACE LIKE HOME "

(2) Now convert the message into numbers, where the space character is 0, and the letters of the alphabet are: A=1, B=2, C=3, ..., Z=26 .

"20,8,5,18,5,0,9,19,0,14,15,0,16,12,1,3,5,0,12,9,11,5,0,8,15,13,5,0,0,0"

(3) Replace each number by the difference between itself and the previous number. If the result is negative, add 27. The first number of course does not change:

"20,15,24,13,14,22,9,10,8,14,1,12,16,23,16,2,2,22,12,24,2,21,22,8,7,25,19,22,0,0"

(4) Divide the numbers into groups of 5, and treat each group as the 5 digits of a number in base 27. The first group of 5 would become for example:

$$20(27)^4 + 15(27)^3 + 24(27)^2 + 13(27) + 14 = 10941926$$

Similarly the whole message becomes:

10941926, 11876369, 779938, 1118634, 1492486, 13676040

(5) Each of the numbers are now multiplied by 5124604

and the remainder found mod 14348891 (a prime number)

"11030774, 5972916, 4155393, 2129744, 10355814, 976860"

(6) They can again be broken down into its 5 digits base 27, (the reverse of step 4) since for example: $11030774 = 20(27)^4 + 20(27)^3 + 11(27)^2 + 10(27) + 5$

And the whole message becomes:

"20,20,11,10,5,11,6,12,8,3,7,22,3,3,12,4,0,5,12,11,19,13,3,13,18,1,22,17,0,0"

(7) The final step before converting these numbers back to letters, is to add to each number the previous number and take the remainder mod 27 (the reverse of step 3)

"20,13,24,7,12,23,2,14,22,25,5,0,3,6,18,22,22,0,12,23,15,1,4,17,8,9,4,21,21,21"

(8) And so the encoded message finally becomes (by reversing step 2):

"TMXGLWBNVYE CFRVV LWOADQHIDUUU"

Decoding this message takes exactly the same steps, except that, instead of multiplying the large numbers by 5124604 and finding the remainder mod 14348891, you must multiply the large numbers by 14 and find the remainder mod 14348891. After all, $5124604 \cdot 14 = 1 \pmod{14348891}$

DATA21.txt (DATA22.txt for the second try) contains 5 lines of data. Each line is less than 251 characters, and may contain blank spaces. Each line represents a message that you must decode using the prime number 14348891 and the multiplicand 14.

Note that there may be trailing spaces, so that each line has a multiple of 5 characters.

Sample input:

```
OSKEZDFEWP KRCLFPMOCPEYPFMXTRXRGULK  
HCCENLVXTXBLTEHKLEARZSRBAYAEY  
GBFKIFHGFMPQJGAEWYGHANKHVUHFXLZDQRYIBZOIRLQDHPBJFD  
NQHFRNGLIDZUERHXG J ZMBYMRNLNLROLERXRZEZFOVJHIYUVYCSBIZ  
LQGGYJXCXLYVYHUPJVVDIFIJSXZMJLBKNNBI
```

Sample output:

```
WHO ARE THE WIZARDS ASKED DOROTHY  
OZ HIMSELF IS THE GREAT WIZARD  
ANSWERED THE WITCH SINKING HER VOICE TO A WHISPER  
HE IS MORE POWERFUL THAN ALL THE REST OF US TOGETHER  
HE LIVES IN THE CITY OF EMERALDS
```

Problem 3 – Bugs

Beetles have infected the trees in one of Canada's oldest parks. These beetles carry the TIK fungus. They travel from one tree to the next and find each new tree in a time that is directly proportional to the square of the distance to that tree. For example, if a tree is 3 meters from a newly infected tree, it would take 9 hours to infect the new tree, and if a given tree is 10 meters from the just infected tree, it would take 100 hours for the bugs to find and infect that tree.

DATA31.txt (DATA32.txt for the second try) contains five sets of data. The first line of each set contains an integer n , between 2 and 100 (not less than 2 and not more than 100) representing the number of trees in the park. The next n lines contain ordered pairs of integers (between 0 and 200) representing the x and y coordinates of the trees in the park. A tree designated as (12, 14) for example is 12 meters east and 14 meters north of the southwest corner of the park. The first tree that is infected is the first tree in the list.

Find the time it takes for the entire park to become infected

Sample input (this data takes up 56 consecutive lines, they are listed here in 5 columns to save space):

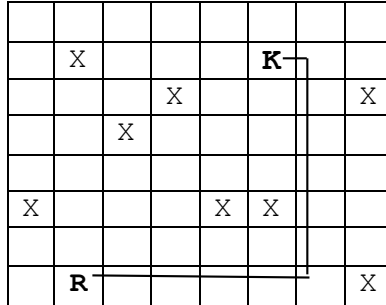
```
2           88 42       61 2         8 32         20 91
190 10      11 56      5 22         41 15        63 67
200 10      69 51      80 75        13           94 102
3           51 77      96 76        41 94        69 56
10 10       43 49      58 28        170 96       3
10 20       97 40      17 7         2 53         67 41
10 15       51 92      86 38        57 60        59 20
30          69 7       76 13        35 36        19 23
8 93        27 87      68 52        123 157
8 25        53 81      98 36        76 8
58 52       28 15      57 38        41 58
17 67       93 18      36 20        23 68
```

Sample output:

```
The whole park is infected after 4 day(s) and 4 hour(s)
The whole park is infected after 2 day(s) and 2 hour(s)
The whole park is infected after 114 day(s) and 6 hour(s)
The whole park is infected after 361 day(s) and 21 hour(s)
The whole park is infected after 88 day(s) and 2 hour(s)
```

Problem 4 – Rook Moves

On an 8 x 8 chess board is found a King, a Rook and various other chess pieces. A Rook may move any number of spaces either horizontally or vertically. In the diagram below, the Rook must take a minimum of 3 steps in order to take the King.



DATA41.txt (DATA42.txt for the second try) contains 40 consecutive lines of 8 characters. Each set of 8 lines represents a distinct chess board, containing one rook, represented by the letter R; one King, represented by the letter K; several other chess pieces, represented by the letter X; and open spaces, represented by the period.

In this exercise you are to find the minimum number of moves for the Rook to take the King.

Sample input

```

.....
.K.....
.....
...X....
.X...X.X
.....
..X..X..
.R.....
.....
.....
.X.....
..XX....
.X...X.X
..XKX...
.XX.X.X.
.RX.....

.....
.K....R.
.....
...X....
.X...X.X
.....
..X..X..
.....
.....
.KX.....
XXXX.XXX
...X....
XXXXXX..
.....X.
..X.X.X.
.R..X...

.....
.K....X.
.....
...X....
.X...XXX
.....
..X..X..
....R...
.....
.KX.....
XXXX.XXX
...X....
XXXXXX..
.....X.
..X.X.X.
.R..X...
```

Sample output

```

The Rook will take the King in 3 move(s)
The Rook will take the King in 6 move(s)
The Rook will take the King in 1 move(s)
The Rook will take the King in 9 move(s)
The Rook will take the King in 2 move(s)
```



ECOO 2006

Programming Contest

Regional Contest

Saturday, April 29 2006

Problem 1 – Split Code

The Spit-Code algorithm is a secret code where each character's ASCII code is first represented by two hexadecimal digits. The following steps are then performed on these digits:

- 1 Starting with the second digit, add the previous digit; if the result is a 2-digit hex number, ignore the high order digit;
- 2 Randomly replace the digit by the corresponding letter in the set: "ABCDEFGHJKLMN" or the set "QRSTUVWXYZKLMNOP"
Where 0 is replaced randomly by either "A" or "Q", 1 is replaced by "B" or "R" etc.

For example: Our school is the best
in hex code: 4F7572207363686F6F6C206973207468652062657374

Then, since $4+F = 13$, the F is replace by 3 (since the high order digit, 1, is ignored)
And since $3+7 = A$, the 7 is replaced by A

And so on: 43AF68AA14AD3B1065B799F8F244BF5D38AA028D47E2
Finally: ETKPWIKKBEKNTLBAGFLXZZPIPSUULPFNTIKKQOSINUXOS

The use of the letters above P is important, so that potential codebreakers do not realize that the code is based on the number 16.

DATA11.txt (DATA12.txt for the second try) contains 5 lines, each of which represent a message which is independently encoded using the Split-Code algorithm. It is your task to write a program that will decode these 5 lines. Each line contains 250 characters or less.

Output should be on a cleared screen or window. Since some messages will wrap and be displayed on more than one line, **separate messages with a blank line.**

Sample Input:

```
ECYXOVHHNPGPRRYMCKAZATVVMAGPVCINPPUXNOESILRJPOQAWOUVLP RRYKQZATJOUCEA
HZPQH LBZPELNPPWKAYODFVMTZIPRYLBWYIOEKZASUULPFNTYKKQYOPVDJNDPFOUCYP
GFLBDDK OUMCHZJPVMOENTULNEHJQCCYVMBXTJOFZPEKPGIPC
GXNLRFXHORYMCRHGMQCSJQGHOSYLRZPYOMCZ
HLBZPUWWMOUFMQXLBNTIKKAPVLNNTLRKREWWMZPQHKBFLAHZLXZZNBXGMKMMRSJOUNEMCRYMSH
```

Sample Output

```
Now by this time Sancho had risen,
rather the worse for the handling
of the friars' muleteers
and stood watching
the battle of his master, Don Quixote
```

Problem 2 – Sum of Squares

Any nonnegative integer can be written as the sum of at most 4 squares:

$$1 = 1^2$$

$$5 = 2^2 + 1^2$$

$$9999 = 99^2 + 14^2 + 1^2 + 1^2$$

Interestingly, it seems that only numbers that have a remainder of 7 when divided by 8, are written as a sum of 4 squares. All others can be expressed as the sum of 3 squares or less.

Write a program that can take a number and express it as the sum of one or more squares. You must use the minimum number of squares possible. For example, 4 can be expressed as $1^2+1^2+1^2+1^2$, however, 4 should be expressed as 2^2 .

You must list the squares in descending order: the largest squares first, the smallest squares last.

In many cases there are more than one solution. You must choose the solution where the first number is as large as possible, and if there is more than one such choice, pick the one where the next number is as large as possible.

For example, there are many solutions for the case of 9999. There are however only four solutions containing 99^2 as one of its terms:

$$9999 = 99^2 + 9^2 + 9^2 + 6^2$$

$$9999 = 99^2 + 10^2 + 7^2 + 7^2$$

$$9999 = 99^2 + 13^2 + 5^2 + 2^2$$

$$9999 = 99^2 + 14^2 + 1^2 + 1^2$$

And in this case, the second term, 14^2 , indicates the proper choice.

DATA21.txt (DATA22.txt for the second try) contains 5 numbers between 1 and 9999 (inclusive). Express each number as a sum of squares, as detailed above.

Sample Input:

```
4
1234
23
456
9999
```

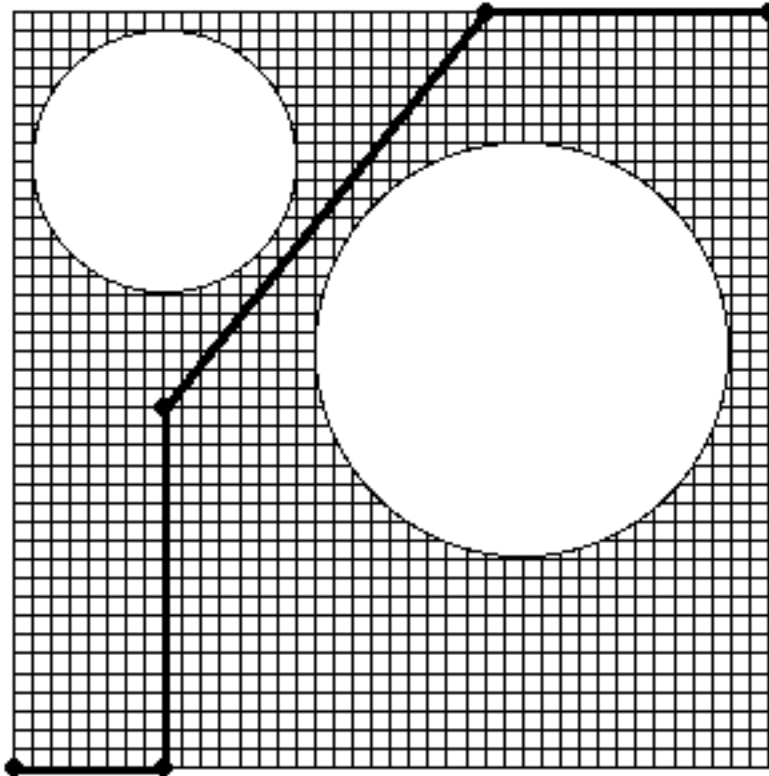
Sample Output

```
4 = 2 squared
1234 = 35 squared + 3 squared
23 = 3 squared + 3 squared + 2 squared + 1 squared
456 = 16 squared + 14 squared + 2 squared
9999 = 99 squared + 14 squared + 1 squared + 1 squared
```

Problem 3 – Polynomial Path

Consider a 40x40 grid, where the lower left corner is (0,0) and the lower right corner is (40, 0) and the upper right (40,40). Within this grid lie two circles, with integer centers, and integer radii. These circles are separated from each other by at least 2 units and neither of them touch any of the four sides of the grid.

Write a program that draws a polynomial path from (0,0) to (40,40) that passes between the two circles. The path may touch the edges of the grid but may not touch the two circles.



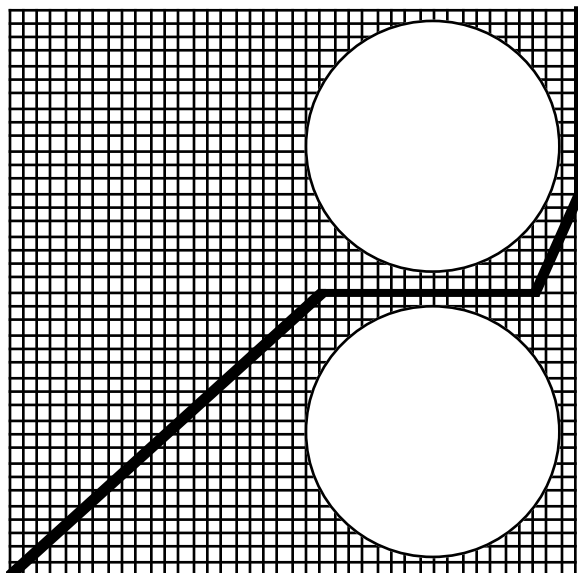
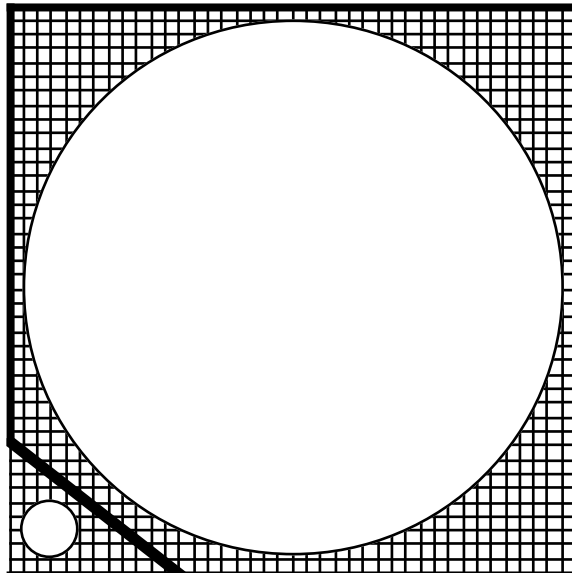
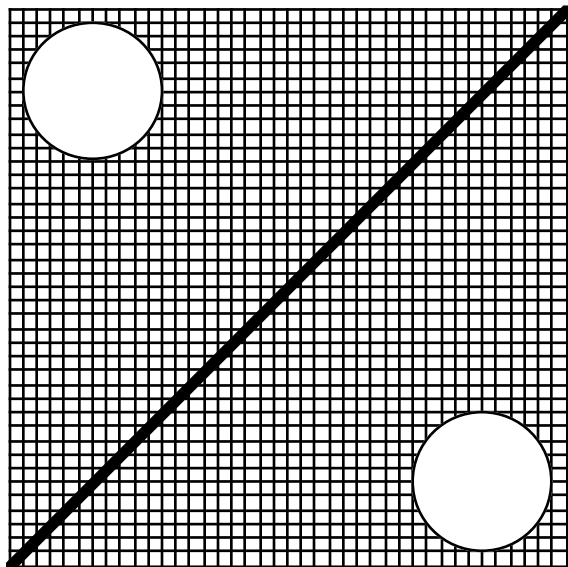
In this example two circles are given, the first one has the center at (8,31) with radius 7, the second circle has its center at (27,22) with radius 11. The path shown starts at (0,0), passes through (8,0), (8,19) and (25,40) and ends at (40,40).

DATA31.txt contains ten lines, representing five sets of two lines. Each line contains 3 positive integers representing the x-coordinate, the y-coordinate and the radius of a circle. Your task is to draw 5 consecutive pictures, each on a separate window or screen, with a prompt, to continue with a keystroke or a mouse click. Each picture must contain a 40x40 grid, the two circles and a path from (0,0) to (40,40) as described.

Sample input (only 3 of 5 examples are shown)

6 34 5
34 6 5
20 20 19
3 3 2
30 30 9
30 10 9

Sample output



Problem 4 – Cryptarithm

A cryptarithm is a mathematical expression in words where each letter represents a digit. The most famous of these is:

SEND		9567
<u>+MORE</u>	Which is uniquely	<u>+1085</u>
MONEY	represented by the sum:	10652

That is: O=0, M=1, Y=2, E=5, N=6, D=7, R=8 and S=9

Write a program that will solve sums of this kind. You may assume for the purpose of this program, that all sums are composed of 3 words, and that the third word (the sum) has one more letter than any of the two addends and, by implication, the first letter of that word represents a 1. Note also, that the first letter of any word cannot represent a 0.

DATA41.txt (DATA42.txt for the second try) contains 5 sets of 3 words on three lines, for a total of 15 lines. The first two words of the set are the addends; the last word of the set of three is the sum. Your output should be formatted as shown below.

The program may consume a lot of time. **Note that only the solutions that are correctly displayed within the first minute of the program will be counted for points.**

Sample Input:

```
AS
A
MOM
DOWN
WWW
ERROR
OOH
FOOD
FIGHT
TAKE
THAT
SHEET
LEAH
LOVES
RUSSIA
```

Sample Output

```
AS      92
A       9
MOM     101
DOWN    9364
WWW     666
ERROR   10030
OOH     8886
FOOD    1883
FIGHT   10769
TAKE    7460
THAT    7547
SHEET   15007

LEAH    9325
LOVES   98437
RUSSIA  107762
```


Problem 1 – Fuel Crisis

Salsa, a fruit salesman, living in Appletown, has his usual route through Banana city, Cabbage town, Dill place, Endive junction, Figville and Grape corners (A,B,C,D,E,F and G for short). Unfortunately the donkey that pulls his cart only runs on carrots, and as long as he can feed it a carrot every kilometer, it will happily pull the cart at 10 kph. Unfortunately, it refuses to work without the carrot, so when our salesman runs out of carrots, he has to hitch the donkey to the back of the wagen and pull the load himself... at 1 kph.

Now the distances in kilometers between the villages is as indicated below:

	<i>Apple town</i>	<i>Banana city</i>	<i>Cabbage town</i>	<i>Dill place</i>	<i>Endive junction</i>	<i>Fig ville</i>	<i>Grape Corners</i>
Apple town	0	28	22	18	32	36	20
Banana city	28	0	17	39	52	40	23
Cabbage town	22	17	0	20	45	40	37
Dill place	18	39	20	0	28	38	46
Endive junction	32	52	45	28	0	22	47
Fig ville	36	40	40	38	22	0	30
Grape Corners	20	23	37	46	47	30	0

In each town of course, Salsa buys all the available supply of carrots, for he hates to pull the wagen himself. It is your task to find the route for Salsa to travel, where he spends the least amount of time on the road. Make sure Salsa is back in Apple town at the end of the trip.

DATA11.txt (DATA12.txt for the second try) contains 5 lines, each containing 7 integers. The 7 integers represents the number of carrots Salsa can buy in each town, in order, from Apple town to Grape Corners. Print out the order of the villages he must visit, and how many hours he spends on the road, to one digit accuracy.

Sample input

```
9 0 50 15 8 5 5
10 10 5 70 80 0 0
30 10 10 50 10 40 0
0 25 15 25 15 35 15
10 20 30 40 30 20 10
```

Sample output

```
The best route is: ACBGFEDA with total time = 77.2 hours
The best route is: ADEFGBCA with total time = 23.2 hours
The best route is: ADEFGBCA with total time = 25.0 hours
The best route is: ADEFGBCA with total time = 43.9 hours
The best route is: ADCBGFEA with total time = 23.4 hours
```

Problem 2 – Blobs

A blob is an arrangement of at least 5 letters of the same kind that are connected. Any two letters are considered connected, if they are neighbours left-right or neighbours up-down. Diagonal neighbours are not considered connected. In the rectangle below, for example, there is a Q-blob, a B-blob and a Y-blob. The group of A's near the upper left corner do not form a blob, because there are only 4 A's that are connected. (the A-group has two diagonal neighbours that are NOT connected.)

```
XAAFZQXBYYVED
PAAPXQQCLYQWZ
AMIAATQQACYFUT
BPSBBQZMKYHLJ
RVBBBBQSNYAPZ
GCIBBHKIZYFPY
```

Write a program that finds the number of blobs and the number of letters in the largest blob. If a particular rectangle is blob-free, indicate that with an appropriate statement, as in the example below.

DATA21.txt (DATA22.txt for the second try) contains 5 sets of data. Each set is composed of a line containing two numbers, x and y followed by x lines of y characters. Note that $4 < x, y < 50$.

Sample Input

```
6 10      13 13      8 7      5 5      7 7
AAXBFUJYAD  NAXUEEQWMBELG  XXBBRRH  SSSSS  ABCDEFG
AAWWANNJYT  NFCFVMTBGNEQG  XKBBRRH  SSSSS  ABCDEFG
KWWWWNNNNN  BXJLYNZXGVOBG  YKKKBHH  SSSSS  ABCDEFG
KKBMEGVCMK  DUWXVCVKNIMOG  BDKKHHH  SSSSS  ABCCCC
KKPPGYYYQQ  HJEETCRMANUGG  MKQDUCI  SSSSS  ABCCDCC
KPPPPKYJQQ  AIEERCCCUOGL  QQONPXV  SSSSS  ABCDDDC
              JMHPYJLGGGGGZ  RFQLPPD  ABCCCC
              RFHLVPDGYYYYY  RFHLVPP
              FVVFOMJGGGGGG
              VQVVVVVOSODWN
              KIHCMOONWQQQ
              IFXXXWOIWMQQS
              QPLXXOOOLWQTK
```

Sample Output

```
There are 4 blobs, the largest contains 7 letters
There are 7 blobs, the largest contains 19 letters
There are 4 blobs, the largest contains 7 letters
There are 1 blobs, the largest contains 25 letters
There are 3 blobs, the largest contains 19 letters
```

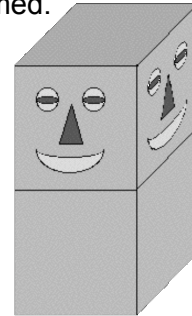
Problem 3 – Tumbledee

Tumbledee is a 2x1x1 block that can tumble in any one of four different directions: North, South, East or West. It can be standing Up or laying Down in any one of six different positions, depending on where its head is in relation to its feet: North, South, East, West (when it is laying down) and Up or Down (depending on when Tumbledee is standing upright or on its head).

Tumbledee stands in the middle of a board like a chess board, except, that the squares extend indefinitely in all four directions. Tumbledee's initial position is on square (0,0). The square immediately to its north is (0,1); immediately to its south is (0,-1), to its east is (1,0) and to its west is square (-1,0). All other squares are numbered consistently, and so for example the square 5 positions to its east and 4 positions to its north is named (5,4).

Tumbledee starts tumbling according to a string of instructions made up of the four letters E,W,N,S indicating the four directions, and the challenge is to keep track of where the head of Tumbledee is located after the series of tumbles it has performed.

For example, the series of tumbles WEEENSS would place its head
W — first at (-2,0), laying down with its head pointing West.
E — then back at (0,0), being again in the Up position,
E — followed by (2,0), pointing East
E — followed by (3,0), now being upside down
N — followed by (3,1), laying down again, pointing South
S — followed by (3,0), again standing on its head
S — followed by (3,-1) with its head pointing North.



DATA31.txt (DATA32.txt for the second try) contains 5 lines of data. Each line represents a set of tumbles and contains no more than 200 letters. Your task is to write a program that will be able to locate Tumbledee's head at the end of each set of tumbles, while assuming that Tumbledee starts out in square (0,0) and in the upright position at the start of each set.

Sample Input:

```
NENNEESWNN
ENEENNNWNNENENNNWSENEEESESWSSES
NEEESENEEESENEEEESWWNEEESWWSSSSWSEENEEESWWN
WNESWSSWWWNNNEEESWWS
E
```

Sample Output

```
Tumbledee's head is at (3,5)
Tumbledee's head is at (12,7)
Tumbledee's head is at (14,-7)
Tumbledee's head is at (-2,-1)
Tumbledee's head is at (2,0)
```

Problem 4 – Word Chain

A well known word game involves changing one word into another word by successively substituting **one letter at a time**, by another letter, in such a way that the intermediate words still make sense.

For example, to change the word ALONE into CRAZY could involve the following steps:

ALONE – CLONE – CRONE – CRANE – CRAZE - CRAZY

Each of these intermediate words are acceptable English words.

Sometimes you are forced to extend the chain to several more words, to ensure that each intermediate word be found in the dictionary.

In this problem we will only consider 5-letter words, and for convenience, there is a “dictionary” of 5-letter words in the input file.

DATA41.txt (DATA42.txt for the second try) contains 10 lines each containing one 5-letter word in capitals, representing 5 sets of two words. The first word is the starting word of the chain, the second the final word. These ten lines are followed by the “dictionary”, composed of an integer n (on line 11), where $30 < n < 50$, followed by n lines, each containing a 5 letter word (in capital letters and in alphabetical order).

Output should be the shortest chain possible, as shown in the sample below. There will always be a solution.

Sample Input (37 lines)

ALONE	26	CRONE	GROWL
CRAZY	ALONE	CROSS	GROWS
CROSS	BLOBS	GLOBE	TRIAL
GLORY	BRAIL	GLOBS	TRISL
GROWL	BRAIN	GLOBS	TRIST
BLOBS	BROIL	GLORE	TRUSL
BRAIN	CLONE	GLORY	TRUST
BROIL	CRANE	GLOSS	
TRIST	CRAZE	GLOWS	
TRIAL	CRAZY	GROSS	

Sample Output

ALONE – CLONE – CRONE – CRANE – CRAZE – CRAZY
CROSS – GROSS – GLOSS – GLOBS – GLOBE – GLORE – GLORY
GROWL – GROWS – GLOWS – GLOBS – BLOBS
BRAIN – BRAIL – BROIL
TRIST – TRISL – TRIAL